PHILIPS

Electronic components and materials

Technical publication

022

User instructions for the serial I/O interface of the MAB8400 series of microcomputers

This publication details the user instructions for the serial input/output (SIO) interface of the MAB8400 series of microcomputers described in an earlier publication*. The software for the various modes of operation is explained with the aid of flowcharts. The hardware is also described in the depth necessary to clarify the software descriptions. A comprehensive description of the hardware of the SIO interface was published in an article entitled 'Serial I/O with the MAB8400 series microcomputers', in EC & A Vol. 3, No. 1, Nov. 1980. The article is available as a separate publication.

SERIAL BUS STRUCTURE

The serial bus shown in Fig. 1 consists of two bidirectional lines; a serial data line (SDA) and a serial clock line (SCL). Pin 2 (SERIAL DATA I/O) of each microcomputer is connected to the SDA line, pin 3 (SCLK) is connected to the SCL line. A pull-up resistor R_p is connected between each line and $+V_{DD}$ so that the open-drain outputs of the microcomputers connected to the bus, together form a wired-AND function via each line. A HIGH level is a logic 1, a LOW level is a logic 0, Each microcomputer can be software programmed to function as a 'transmitter' or a 'receiver' operating in either a 'master' or a 'slave' mode.

SERIAL DATA TRANSFER

Figure 2 shows the serial data transfer sequence for the SIO interface. The following conditions can be distinguished:

F (free): The bus is free; data line SDA and clock line SCL are both HIGH.

S (start): A data transfer commences with a 'start' condition during which the level of data line SDA changes from HIGH to LOW, and clock line SCL remains HIGH. The bus is now busy.

C (change): During the LOW period of the clock, the data bit to be transmitted is applied to data line SDA. The level on the SDA line may therefore change during this period.

D (data): One data bit is transmitted during the HIGH period of the clock. The level (bit state) on line SDA must remain stable during this period to prevent it being interpreted as a 'start' or 'stop' condition.

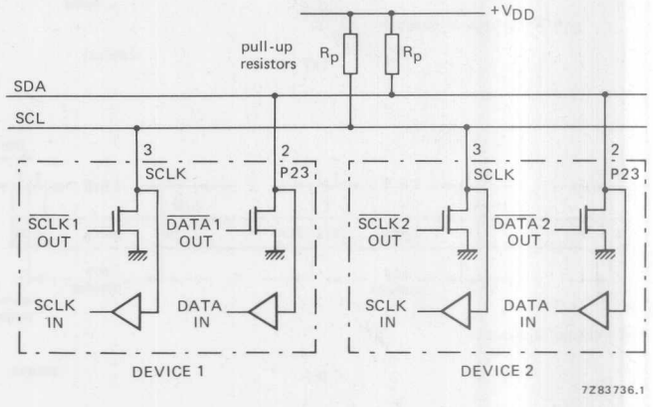


Fig. 1 Connection of two microcomputers to the serial bus.

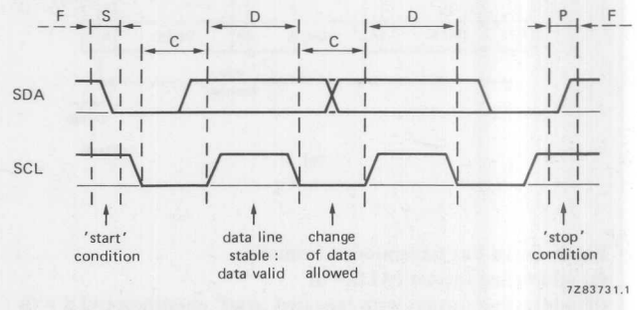


Fig. 2 Sequence of bit transfer on the serial bus.

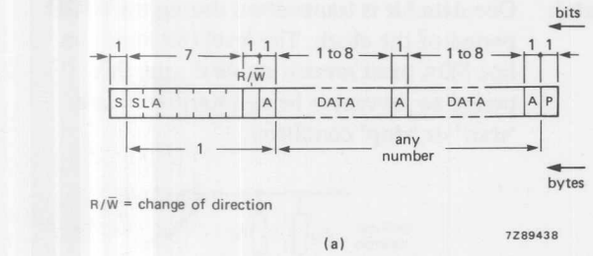
* 'MAB8400 family of single-chip 8-bit microcomputers'.

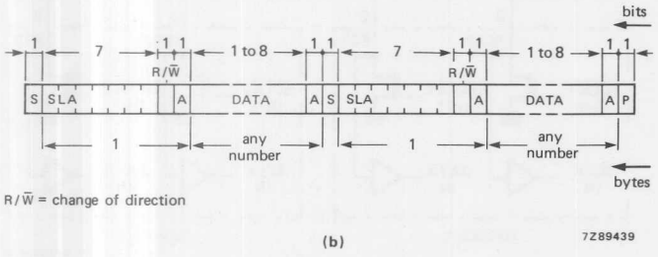
P (stop): A data transfer concludes with a 'stop' condition during which the level on data line SDA changes from LOW to HIGH and clock line SCL remains HIGH. The bus is now free again.

SERIAL DATA FORMATS

The three data formats that can be serviced by the serial I/O are shown in Fig. 3. The first byte (SLA) after a 'start' condition (S) always consists of 8 bits. By programming an internal flag, an acknowledge bit (A) may be inserted after each byte.

In the first two formats (addressing formats) byte SLA is the address of the 'slave' to be selected by a 'master'. The least-significant bit R/$\overline{W}$ of the address byte indicates the direction of transmission of the following data bytes. If R/$\overline{W}$ is LOW, the 'master' will write (transmit) data into the selected 'slave'; if it is HIGH, the 'master' will read (receive) data out of the 'slave'. The format in fig. 3(b) is used when the direction of a transfer has to be changed during a transmission. In the free data format, the direction of transmission remains constant throughout the data transfer.





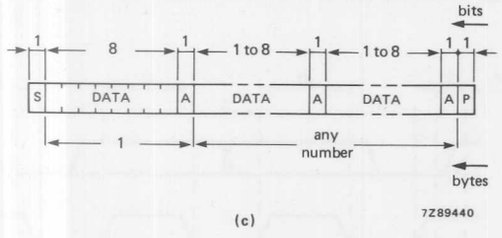


Fig. 3 Serial bus transmission formats;
(a) addressing format (ALS = 0)
(b) addressing format with repeated 'start' condition (ALS = 0)
(c) free data format (ALS = 1).

OPERATING MODES OF THE SERIAL I/O INTERFACE

The serial I/O can function in the following four basic operating modes to service all facilities of the I^2C-bus, P/C-bus or point to point connections:

- 'master transmitter'
- 'master receiver'
- 'slave receiver'
- 'slave transmitter'.

Master transmitter

In this mode, data assembled in one of the previously described data formats is shifted-out on SDA in synchronism with the self-generated clock pulses on SCL. The clock pulses are inhibited and SCL held LOW when the intervention of the processor is required after a byte has been transmitted.

Master receiver

This mode can only be entered from the 'master transmitter' mode. With either of the address formats (Fig. 3a or 3b), the 'master receiver' mode is entered after address byte SLA and bit R/$\overline{W}$ have been transmitted if R/$\overline{W}$ was HIGH. Serial data bits received on bus line SDA by a 'master receiver' are shifted-in in synchronism with the self-generated clock pulses on SCL. The clock pulses are inhibited and SCL held LOW when the intervention of the processor is required after reception of a byte. At the end of a transfer, the 'master receiver' generates the 'stop' condition P.

Slave receiver

Serial data bits received on bus line SDA by a 'slave receiver' are shifted-in in synchronism with the clock pulses at SCL which are generated by the 'master' device. A 'slave receiver' does not generate clock pulses, it only holds clock line SCL LOW whilst intervention of the processor is required following the reception of a byte. 'Start' and 'stop' conditions are recognized and interpreted accordingly.

Slave transmitter

The 'slave transmitter' mode can only be entered from the 'slave receiver' mode. With either of the address formats (Fig. 3a or 3b), the 'slave transmitter' mode is entered if the address received in byte SLA corresponds which the own slave address and the R/$\overline{W}$ bit is HIGH. A 'slave transmitter' shifts the serial data out on data line SDA in synchronism with the clack pulses which are generated on clock line SCL by the 'master' device. A 'slave transmitter' does not generate clock pulses, but only holds clock line SCL LOW whilst intervention of the processor is required after transmission of a byte. 'Start' and 'stop' conditions are recognized and interpreted accordingly.

CLOCK GENERATOR SYNCHRONIZATION

In an I^2C bus system, more than one device may simultaneously generate clock pulses of different repetition rates. All the clock pulse generators are therefore synchronized with the pulses on clock line SCL. The clock pulse synchronization sequence is illustrated in Fig. 4.

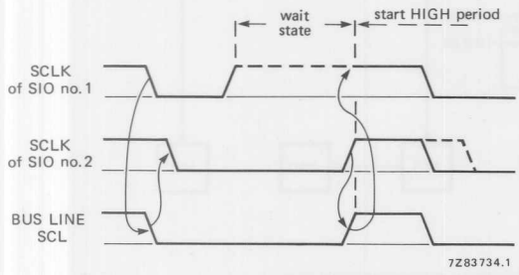


Fig. 4 Synchronization of two SIO clock generators.

The wired-AND property of the SCLK outputs of the devices connected to clock line SCL means that the first device to start generation of a LOW clock period overrules all others. When this HIGH to LOW transition occurs, on clock line SCL, the clock generators of the other devices are forced to start generation of their own LOW clock period. The time for which clock line SCL is held LOW is determined by the device with the longest LOW period and all the other devices must wait until the clock line goes HIGH again. By this method, clock pulse synchronisation is obtained wherein the device which generates the highest frequency clock pulses determines the duration of the HIGH periods on clock line SCL, and the device which generates the lowest frequency clock pulses determines the duration of the LOW periods.

ARBITRATION PROCEDURE

In an I^2C bus system, more than one device may simultaneously start a data transmission on data line SDA. The abitration procedure illustrated in Fig. 5 is therefore used to prevent the data on bus line SDA from being corrupted by a simultaneous data transmission from another device.

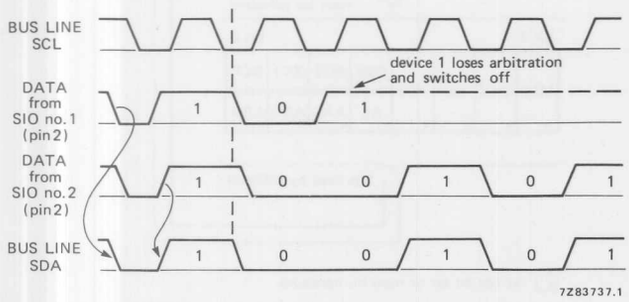


Fig. 5 Arbitration between two devices that simultaneously start to transmit.

Due to the wired-AND property of the SERIAL DATA outputs of devices connected to data line SDA, a HIGH level on SDA is overruled by a LOW level from another device. If this occurs, the output stage of the device that has generated the HIGH level is switched off so that data line SDA is surrendered to the device that has generated the LOW level. The clock output stage is switched off after the serial data byte on the bus line has been completed. The device that is first to transmit a 0 data bit (the byte with the lowest binary value) will therefore win the arbitration. To prevent incorrect interpretation by the 'transmitter', the arbitration procedure is inactive during an acknowledge bit.

CIRCUIT DESCRIPTION AND OPERATION

The block diagram of the SIO interface given in Fig. 6 shows that the internal microcomputer bus communicates with the following four registers:

- data shift register S0
- address register S0'
- status register S1
- clock control register S2.

Eight instructions are available so that these four registers can be written into or read via the internal microcomputer bus. All the bits in the registers, except the PIN (pending interrupt not) bit in status register S1 are set to 0 by a RESET. The PIN bit is set to 1 by a RESET.

The address comparator shows in Fig. 6 can instruct the interrupt logic block (if enabled) to generate an SIO interrupt request to the microcomputer.

Communications with the four registers and the function of the address comparator will now be described in more detail.

Data shift register S0

To address this 8-bit shift register, the ESO (enable serial output) bit in status register S1 must be a 1. The write instruction MOV S0, A or MOV S0, # data causes data register S0 to be parallel-loaded via the internal bus with the data for transmission. During transmission, the contents of the register are shifted out, most significant bit first, onto data bus line SDA.

During reception, the first (most-significant) bit of the data byte which is received on data line SDA is shifted into the least-significant bit position in the register. As each subsequent bit is received, the register contents are shifted one position to the left. In the acknowledgement mode, acknowledgement bit ACK is not shifted into register S0 but into the LRB (last received bit) position of S1. The read instruction MOV A, S0 causes the contents of register S0 to be parallel-loaded into the accumulator via the internal bus.

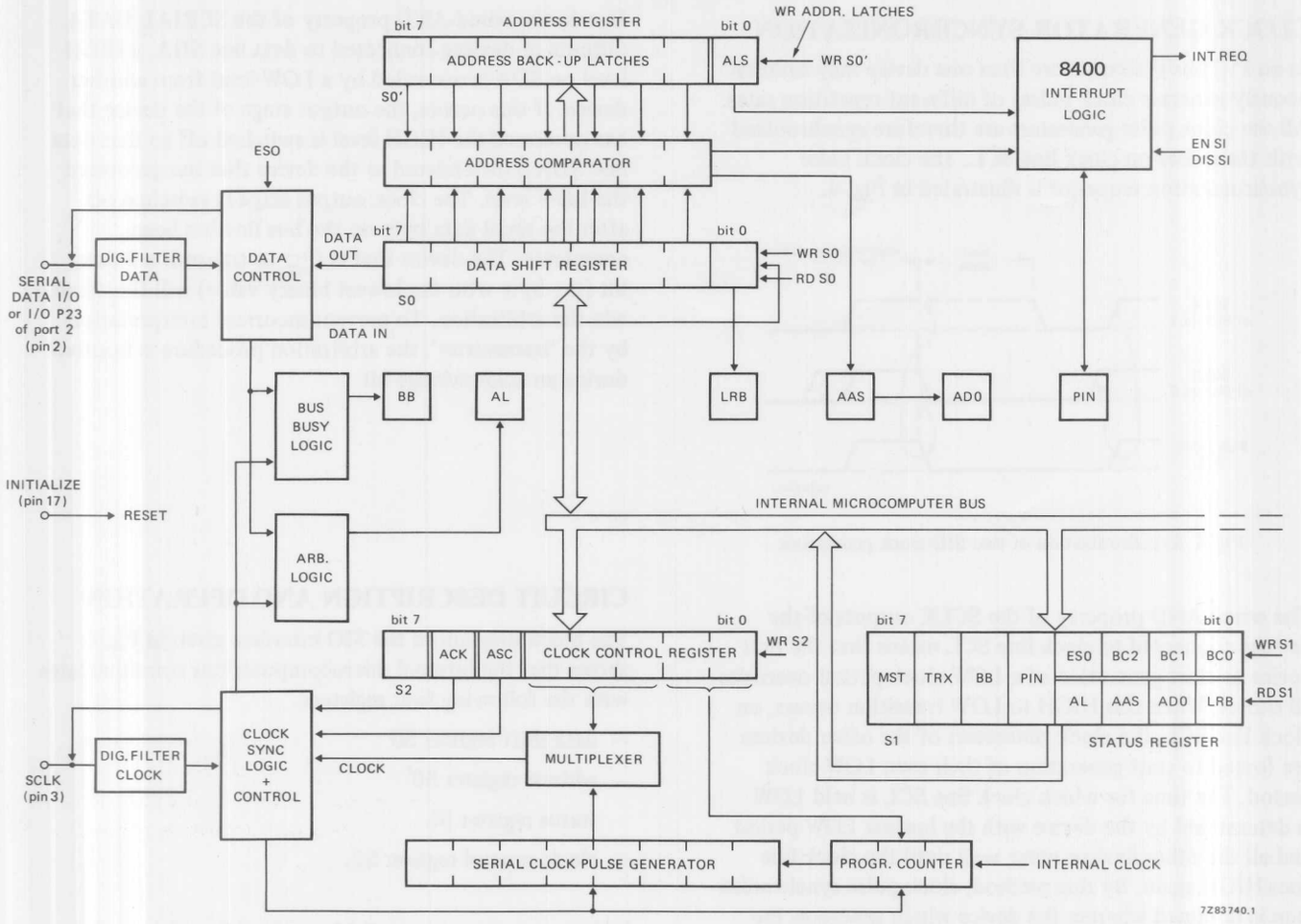


Fig. 6 Block diagram of the MAB8400 SIO interface.

Address register S0'

The seven most-significant bits in this 8-bit register are the slave address. The least significant bit ALS (always selected) denotes the addressing format (Fig. 3a or 3b) if it is 0, or the free data format (Fig. 3c) if it is 1. To address register S0', the ESO bit in status register S1 must be 0. The write instruction MOV SO, A or MOV S0, # data causes the address register to be parallel-loaded via the internal bus.

Address comparator

This comparator activates the interrupt logic if the received address in data shift register S0 is the same as the slave address in address register S0'. The interrupt logic is also activated if all seven bits of the received address, plus the R/$\overline{W}$ direction bit, are 0.

Status register S1

The status word in status register S1 is shown in Fig. 7. It contains all information regarding the status of the SIO interface. Read instruction MOV A, S1 causes the status word to be read out via the internal bus. To control the SIO interface, information is written into the status register with the write instruction MOV S1, A or MOV S1, # data. There are two latches at the four least-significant bit positions in the status register so that two bits can be stored at each position. The four least-significant bits ESO, BC2, BC1, and BC0 are control bits that can only be written. The four least-significant bits AL, AAS, AD0 and LRB are status bits that can only be read. The four most-significant bits are MST, TRX, BB and PIN which can be either written or read. The function of each bit in the status register will now be described.

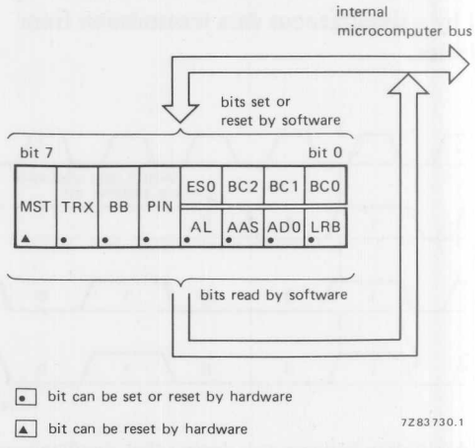


Fig. 7 Bit allocation of the status word in register S1.

MST: 'master' bit. If this bit is 1, the SIO interface is in the 'master' mode and it generates the clock pulses on clock line SCL for timing the transmission or reception of the serial data. If the MST bit is 0, the SIO interface is in the 'slave' mode and the clock pulses are received from the 'master' device on clock line SCL. The MST bit is set to 0 by the hardware if an arbitration is lost. When a data transfer has been completed, a 'stop' condition is generated by the 'master' device and the hardware resets the MST bit to 0.

TRX: 'transmitter' bit. If this bit is 1, the SIO interface is in the 'transmitter' mode and the data in register S0 is shifted out on data line SDA in synchronism with the pulses on clock line SCL. If TRX is 0, the SIO interface is in the 'receiver' mode and the data on bus line SDA is shifted into data register S0 in synchronism with the pulses on clock line SCL. The TRX bit is set to 0 by the hardware if an arbitration is lost. When a data transfer has been completed, a 'stop' condition is generated by the 'master' device and the hardware resets the TRX bit to 0. In either of the addressing formats (Fig. 3a or 3b), the state of the TRX bit is set by the hardware in accordance with the state of read/write direction bit R/$\overline{W}$. The operating modes set by the MST and TRX bits of the status word are summarized in Table 1.

TABLE 1 Operating modes set by bits MST and TRX

MST	TRX	operating code
0	0	'slave receiver'
1	0	'master receiver'
0	1	'slave transmitter'
1	1	'master transmitter'

BB: 'bus busy' bit. This bit indicates the state of the serial bus as determined by the busy logic in the 'slave' mode. If it is 0, the bus is free. If it is 1, the bus is occupied. On reception of a 'start' condition, the bus busy logic sets BB to 1. BB is reset to 0 one LOW period of the internal serial clock after reception of a 'stop' condition. In the 'master' mode, BB is controlled by software. To start a transmission with a 'start' condition, bits MST, TRX and BB of the status word are set to 1. To finish a transmission with a 'stop' condition, bit BB is set to 0 and bits MST and TRX are set to 1.

PIN: 'pending interrupt not' bit. Setting this bit to 0 initiates a serial I/O interrupt if the EN SI (enable serial interrupt) instruction has been previously carried out. As long as PIN is 0, the clock pulses are inhibited and clock line SCL is held LOW. The pin bit is only set to 0 when:

- A complete byte has been transmitted (even if arbitration has been lost).
- The address comparator has recognized its own slave address in one of the addressing formats shown in Fig. 3(a) or 3(b), or the address of all (8) zeros has been received (ALS flag = 0).
- Another byte has been received after an address has been previously recognized.
- A byte has been received in the free data format shown in Fig. 3(c) (ALS flag = 1).

The PIN bit is reset to 1 by:

- Read instruction MOV A, S0 or write instruction MOV S0, A or MOV S0, # data.
- Loading a 1 into the PIN bit position in status register S1 as a result of write instruction MOV S1, A or MOV S1, # data.

ESO: 'enable serial output' bit. When this bit is 1, the SIO interface is enabled and can receive or transmit on serial data line SDA, in synchronism with the pulses on clock line SCL. When the ESO bit is 0, the SIO interface is disabled, pin 2 reverts to its P23 (I/O line of port 2) function and pin 3 (SCLK) is in the high impedance state. Whilst ESO is 0, a 7-bit slave address, plus ALS bit, can be loaded into address register S0′ with write instruction MOV,S0, A or MOV S0, # data. As long as ESO remains 0, PIN is held HIGH, AL is held LOW and the contents of data shift register S0 are not affected.

BC2, BC1 and BC0: 'bit count'. As shown in Table 2, the binary-coded number preset in bit positions BC2, BC1 and BC0 is the number of bits (excluding the acknowledge bit) of the next byte which are yet to be received or transmitted.

TABLE 2 Binary numbers in bit-count locations BC2, BC1 and BC0.

BC2	BC1	BC0	bits/byte without ACK	bits/byte with ACK
0	0	1	1	2
0	1	0	2	3
0	1	1	3	4
1	0	0	4	5
1	0	1	5	6
1	1	0	6	7
1	1	1	7	8
0	0	0	8	9

A 'start' condition resets the bit count to 000, so the first byte to be received or transmitted will always consist of eight bits excluding the acknowledge bit. After each complete byte has been received or transmitted, the bit count has reduced to 000.

AL: 'arbitration lost' bit. This bit is set to 1 when the arbitration logic sets the AL flag because the SIO interface has lost an arbitration when it is in the 'master transmitter' mode. It is also set to 1 if an attempt is made to occupy the bus at the same moment that S1 is written. After a byte has been transmitted, PIN is set to 0 and the status word in register S1 indicates in which mode the SIO has been addressed. AL is reset to 0 by read instruction MOV A, S0 or any of the write instructions MOV S0, A MOV S0, # data MOV S1, A or MOV S1, # data.

AAS: 'addressed as slave' bit. This bit is set to 1 when the address comparator sets the AAS flag because it has recognized its own slave address or an address of all (8) zeros. The AAS flag is also set if the first byte has been received in the free data format (ALS = 1). The AAS bit is reset to 0 by the read instruction MOV A, S0 or write instruction MOV A, S0 or MOV S0, # data.

AD0: "address zero' bit. This bit is set to 1 if the address comparator detects the address of all (8) zeros. The AD0 bit is reset to when a 'start' or 'stop' condition is detected.

LRB: 'last received bit'. If a byte is transmitted with an acknowledge (ACK) bit, the LRB bit position in status register S1 of the 'transmitter' contains the acknowledgement of the 'receiver'. When LRB is 1, the 'receiver' has not acknowledged reception. When LRB is 0, reception of the transmission has been acknowledged. If a transmission does not include an acknowledgement bit, the last bit of the transmitted or received byte will be in the LRB position.

Clock control register S2

This is an 8-bit write-only register with bit positions as shown in Fig. 8. The seven bits S20 to S26 can be parallel-loaded into the register with write instruction MOV S2, A or MOV S2, #data. The eighth bit position (S27) is not used. Bits S20 to S24 control the frequency of the clock pulses for the SIO interface. Bit S25 (ASC) determines whether or not the mark/space ratio of the clock pulses is 1 : 1. Bit S26 (ACK) determines whether or not the device is operating in the acknowledgement mode.

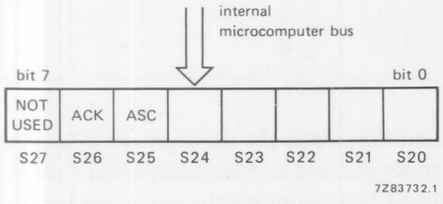


Fig. 8 Bit allocation in clock control register S2.

S20 to S24: clock frequency control bits. These bits form a binary-coded frequency control number (0 to 31) which determines the factor by which the crystal-controlled frequency of the serial clock-pulse generator will be divided to obtain the required frequency of pulses on clock line SCL. Table 3 lists the divisors together with hexadecimal equivalents of the frequency control numbers, and shows the resulting frequency of the clock pulses on SCL if the clock pulse generator is controlled by a 4,43 MHz crystal.

ASC: 'asymmetrical clock' bit at S23. If this bit is 1, the clock pulses on SCL have a mark/space (HIGH to LOW) ratio of 3 : 1 and the binary-coded clock frequency control number may only be HEX'18', HEX'19', HEX'1A' or HEX'1B' which gives a clock-pulse frequency range of 1,4 kHz to 2,3 kHz. This is the low-speed mode of the SIO interface. For example, if the clock-frequency control number HEX'19' is in bit positions S20 to S24, the frequency of the pulses on clock line SCL will be about 1,9 kHz (1/f = 520 μs). Since ASC is 1, the mark/space ratio will be 3 : 1 so that the HIGH time of the clock pulses is 390 μs and the LOW time is 130 μs.

If ASC is 0, the pulses on clock line SCL have a mark/space ratio of 1 : 1 and any of the clock-pulse frequency control numbers, except 0, may be loaded into bit positions S20 to S24. The clock-pulse frequency may therefore be selected within the range 700 Hz to 114 kHz. This is the high-speed mode of the SIO interface.

TABLE 3 Clock pulse frequency control when using a 4,43 MHz crystal.

HEX. S20-S24 code	divisor	approx. f_{clock} (kHz)
0	NOT ALLOWED	
1	39	114
2	45	98
3	51	87
4	63	70
5	75	59
6	87	51
7	99	45
8	123	36
9	147	30
A	171	26
B	195	23
C	243	18
D	291	15
E	339	13
F	387	11
10	483	9,2
11	579	7,7
12	675	6,6
13	771	5,8
14	963	4,6
15	1155	3,8
16	1347	3,3
17	1539	2,9
18*	1923	2,3
19*	2307	1,9
1A*	2691	1,7
1B*	3075	1,4
1C	3843	1,2
1D	4611	1,0
1E	5379	0,8
1F	6147	0,7

* only values that may be used in the low speed mode (ASC = 1).

ACK: 'acknowledge' bit at S26. For operation in the acknowledgement mode, the ACK bit position in register S2 must be loaded with a 1. The acknowledgement procedure of the SIO interface is illustrated in Fig. 9. Its operation for devices in each of the four SIO interface modes will now be described.

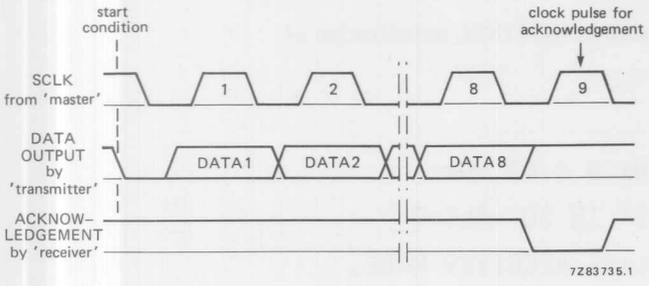


Fig. 9 Acknowledgement between 'receiver' and 'transmitter'.

If the ACK bit in register S2 of a 'master transmitter' is 1, the device generates an extra pulse on clock line SCL at the end of each transmitted byte. During this clock pulse, the SERIAL DATA I/O (pin 2) becomes an input and, if a 'receiver' generates an acknowledgement (LOW level on data line SDA), it resets the LRB flag to 0. If an acknowledgement is not received, the level on data line SDA remains HIGH and this sets the LRB flag to 1.

A 'slave transmitter' reacts to an acknowledgement in the same manner as a 'master transmitter' except it receives the extra clock pulse from the 'master' device instead of generating it.

If the ACK hit in clock control register S2 of a 'master receiver' is 1, the device generates an extra clock pulse after each received byte and applies it to clock line SCL. During this clock pulse, the SERIAL DATA I/O (pin 2) becomes an output where a LOW level is generated on data line SDA to acknowledge receipt of the transmission.

A 'slave receiver' generates an acknowledgement in the same manner as a 'master receiver' except that it receives the extra clock pulse from the 'master' device instead of generating it.

It should be noted that, if a device is changed from a 'receiver' to a 'transmitter' after the R/$\overline{W}$ bit, an acknowledgement will be generated for the last byte that was received. If a device is changed from a 'transmitter' to a 'receiver' after the R/$\overline{W}$ bit, the acknowledgement will be accepted for the last byte that was transmitted.

If the ACK bit position in clock control register S2 is loaded with a 0, the SIO interface is no longer in the acknowledgement mode. A 'master' device does not then generate an extra clock pulse after each byte and a 'receiver' does not generate an acknowledgement LOW level on bus line SDA.

PROGRAMMING

1. State after RESET

A reset (HIGH to pin 17 of the microcomputer) clears all the SIO interface registers (S0, S0′, S1 and S2) except for the PIN (pending interrupt not) bit in status register S1 which is set to 1. Because the ESO (enable serial output) bit in status register S1 is 0, the SIO interface is disabled, pin 2 reverts to its I/O function P23 of port 2 (HIGH after the RESET) and pin 3 (SCLK) is in the high impedance state. An initialization procedure must therefore be carried out before the SIO interface can be used to transfer serial data.

2. Initialization procedure

A flow chart of the initialization procedure is given in Fig. 11. Each step of the procedure will now be explained in more detail.

2.1 Clock control register S2. The first step of the initialization procedure is to load register S2. Bit position S20 to S24 are loaded, as indicated in Table 3, to set the frequency of the pulses on clock line SCL. Bit positions ASC and ACK are loaded as previously described to determine whether the SIO interface operates in the low-speed or high-speed mode, and whether or not it operates in the acknowledgement mode. After S2 has been loaded, the clock pulse generator requires one LOW period of its programmed clock cycle-time to stabilize. Since a data transfer must not be started during this period, the BB flag is set to indicate that the serial bus is busy. The BB flag is reset when the clock generator has stabilized.

2.2 Address register S0′. The next step is to load address register S0′. To address this register, the SIO interface must be disabled by leaving the ESO (enable serial output) bit in status register S1 at 0. Address register S0′ is loaded with the 7-bit slave address plus the ALS bit. If the ALS bit is 1, the message is in the free data format so the slave address in S0′ is immaterial.

2.3 Status register S1. The final step of the initialization procedure is to change the ESO bit in status register S1 to 1. All the other bits in register S1 remain as they were following the RESET, i.e. all bits 0 except PIN which is 1. The status register therefore contains HEX'18' and the SIO interface is in the 'slave receiver' mode and ready to receive serial data.

2.4 Enable/disable serial I/O interrupt. The instruction EN SI must be performed to enable the serial interrupt logic. If the serial interrupt logic is not enabled, the SIO interface can be serviced by polling PIN.

2.5 Example of an initialization procedure. After a RESET and a jump to label INSI, initialization of the SIO interface can be achieved as follows:

```
INSI   MOV S2, #H'43'    WITH ACK, FSCLK≃87 kHz @ 4.43 MHz
       MOV S0, #H'84'    LOAD SLAVE ADD HEX'42' IN S0' ALS=0
       MOV S1, #H'18'    ENABLE SIO, SELECT SLAVE RECEIVER MODE.
       EN SI             ENABLE SERIAL I/O INTERRUPT
```

TABLE 4 Possible bit combinations of the status word.

serial I/O status word S1								mode	concerned received or transmitted serial byte	software response
MST	TRX	BB	PIN	AL	AAS	AD0	LRB			
0	0	1	0	0	1	1/0	X	SLV/REC	own slave/all 0's address received with R/$\overline{W}$ = 0	write 00111XXX to S1 or a dummy read of S0
0	0	1	0	0	0	1/0	X	SLV/REC	data received in S0	00101XXX to S1 and/or read S0
0	1	1	0	0	1	0	X	SLV/TRX	own slave/all 0's address received with R/$\overline{W}$ = 1	write 01101XXX to S1 and/or load S0
0	1	1	0	0	0	0	1/0	SLV/TRX	transmitted out S0. LRB: 1/0 → NO ACK/ACK retruned	write 01101XXX to S1 and/or load S0, or write 00111000 to S1 (end)
1	1	1	0	0	0	0	1/0	MST/TRX	transmitted out S0. LRB = 1/0 → NO ACK/ACK returned	write 11101XXX to S1 and/or load S0, or write 11011000 to S1(STO.C)
1	0	1	0	0	0	0	1/0	MST/REC	slave address with R/$\overline{W}$ = 1 transmitted LRB = 1/0 → NO ACK/ACK returned	write 10111XXX to S1 or a dummy read of S0
1	0	1	0	0	0	0	X	MST/REC	data received in S0	write 10101XXX to S1 and/or read S0, or write 11011000 to S1(STO.C)
0	0	1	0	1	0	0	X	MST/TRX has lost arbitration	slave address or data transmitted and arbitration lost	a dummy read of S0
0	0	1	0	1	1	0	X	MST/TRX has lost arbitration now SLV/REC	during transmitting of slave address arbitration lost and own slave/all 0's address received with R/$\overline{W}$ = 0	write 00111XXX to S1 or a dummy read of S0
0	1	1	0	1	1	0	X	MST/TRX has lost arbitration now SLV/TRX	during transmitting of slave address arbitration lost and own slave/all 0's address received with R/$\overline{W}$ = 1	write 01101XXX to S1 and/or load S0
0	0	0	1	0	0	0	X		the bus is free, a transmission may be started	
X	X	1	X	X	X	X	X		the bus is busy, a transmission may not be started	
X	X	X	1	X	X	X	X		no serial I/O interrupt pending i.e. no complete byte received or transmitted (polling PIN)	

X = 1 or 0, not relevant for status.
Underlined bits not relevant for status.

XXX = BC2, BC1, BC0 according to table 2.
(STO.C) = generation of the STOP CONDITION.

The SIO interface is now in the 'slave receiver' mode. If address HEX'42' is received, the device will generate an acknowledge bit and cause a serial I/O interrupt call. Further software responses depend on the contents of status register S1 as shown in Table 4.

3. Programming the SIO interface to transmit data

After completion of the initialization procedure, serial data can be transmitted by selecting the 'master transmitter' mode as shown by the flow chart in Fig. 12. If the device is connected in multi-transmitter bus system, the state of the BB flag must be tested to verify whether the serial bus is free. If the bus is free (BB = 0), data register S0 is loaded with the first byte for transmission. Bit positions MST, TRX, and BB in status register S1 must each be loaded with a 1 to immediately start the data transmission beginning with a 'start' condition. When the first byte has been transmitted, the PIN flag is reset to 0 and a serial interrupt call is initiated if the instruction EN SI has been performed. The status word in register S1 determines the software responses that will ensue as listed in Table 4. An example of a program which generates the 'start' condition and transmits the contents of working register R_r is as follows:

```
MSTX    MOV A,S1          LOAD STATUS WORD TO ACCU
        JB5 MSTX          TEST IF BUS IS FREE
        MOV A,Rr          LOAD BYTE TO BE TRANSMITTED FROM Rr
;                         TO ACCU
        MOV SØ,A          LOAD BYTE TO BE TRANSMITTED FROM ACCU
;                         TO SØ
        MOV S1,#H'F8'     SELECT MASTER TRANSMITTER MODE AND
;                         START TRANSMISSION
```

After checking that the serial bus is free, but before the transmission starts, another device may engage the bus. If this occurs, the SIO interface will not start its transmission. To prevent received data from being corrupted, the SIO hardware inhibits data from being written into data register S0 and status register S1 by software. If a software instruction attempts to write into data register S1, the AL (arbitration lost) flag is set to indicate that the attempt to transmit has failed. When the AL flag is set, a serial interrupt request is always generated at the end of the serial byte.

4. Software responses after transmission or reception of a byte

After transmission or reception of a byte, the PIN flag is reset to 0. If a serial I/O interrupt call to program memory location 5 (serial interrupt enabled) is made or if the PIN flag is polled (serial interrupt disabled, a software response has to be initiated to service the SIO interface. As long as the PIN flag is 0, transfer of serial data is halted because clock bus line SCL is held LOW. Reading out or writing information to data shift register S0 sets the PIN flag to 1 and allows the transfer of serial data to continue.

The status word in register S1 contains all the information to determine the required software response. It also indicates the operating mode of the SIO interface, and gives information about the transmitted or received serial byte. Table 4 lists all possible bit combinations for the status word. It must be remembered that the four least-significant bits of the status word that can be read are not the same as those that can be written (see Fig. 7). The least-significant bit (LRB) of the status word during a read operation only indicates an acknowledgement if the SIO interface is programmed to operate in the acknowledgement mode i.e. the ACK bit in clock control register S2 is set to 1. If the ACK bit is 0, the LRB bit position in the status register contains the last bit of the byte that has been transmitted or received.

The following is an example of software response by a 'master transmitter'. The next 8-bit byte to be transmitted is in working register R_q.

```
        ORG H'Ø5'       SERIAL INTERRUPT VECTOR
        JMP SIR         JUMP TO SERIAL INTERFACE ROUTINE
*
SIR     SEL RB1         SELECT REGISTER BANK 1
        MOV Rp,A        SAVE ACCU CONTENTS IN Rp
        MOV A,S1        LOAD SERIAL STATUS IN ACCU
        JB7 TXTS        TEST MST BIT
        JMP SLVR        JUMP TO SLAVE ROUTINE IF MST=Ø
TXTS    JB6 TNBY        TEST TRX BIT
        JMP MRRC        JUMP TO MST/REC ROUTINE IF TRX=Ø
TNBY    MOV A,Rq        LOAD NEXT BYTE TO BE TRANSMITTED
*                       FROM Rq TO ACCU
        MOV SØ,A        LOAD IT TO SØ=START TRANSMISSION
        MOV A,Rp        RESTORE CONTENTS OF ACCU
        RETR            RETURN TO MAIN PROGRAM
```

Note that bit counter bits BC0, BC1 and BC2 in the status register need not be preset as they can all remain 0 because 8 data bits have to be transmitted (see Table 2). The state of the LRB flag can be tested to check whether the transmitted data has been acknowledged. The result must be interpreted by the programmer.

5. Generation of the 'stop' condition

A data transfer ends with a 'stop' condition generated by the 'master' device. To generate the 'stop' condition, the program in section 4 is followed until TXTS. After checking that the last byte has been transferred, the program continues with:

```
        MOV S1,#H'D8'   RESET BB TO Ø, MST, TRX, PIN TO 1.
*                       THIS GENERATES THE STOP CONDITION
        MOV A,Rp        RESTORE CONTENTS OF ACCU
        RETR            RETURN TO MAIN PROGRAM
```

6. Generation of a repeated 'start' condition

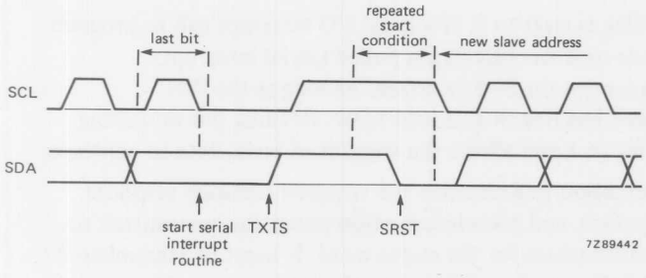


Fig. 10 Repeated 'start' condition on the serial bus.

Figure 10 shows the generation of a repeated 'start' condition followed by a new slave address. The addressing format with repeated 'start' condition is shown in Fig. 3(b). The serial bus remains busy at the end of the preceding data transfer because a 'stop' condition is not generated. The repeated 'start' condition is generated by a program which follows the program in section 4 until TXTS. After a check to verify that a repeated 'start' condition must be transmitted, the program continues with:

```
        MOV S1,#H'18'   RESET MST, SLV, BB TO Ø
¥                       SET PIN TO 1
¥                       NOW BOTH SDA AND SCL
¥                       BECOME HIGH (NO STOP CONDITION)
MSTX    MOV A,S1        LOAD STATUS WORD TO ACCU  **
        JB5 MSTX        TEST BB. AS LONG AS THE INTERNAL SERIAL
¥                       CLOCK IS LOW, BB = 1      **
        MOV A,Rq        LOAD NEW SLAVE ADD. TO ACCU
        MOV SØ,A        LOAD NEW SLAVE ADD. TO SØ
SRST    MOV S1,#H'F8'   SELECT MASTER TRANSMITTER MODE
¥                       START TRANSMISSION
        MOV A,Rp        RESTORE CONTENTS OF ACCU
        RETR            RETURN TO MAIN PROGRAM
```

** These instructions may be omitted if it is certain that the LOW period of the serial clock finishes before label SRST is reached

7. Generation of the 'stop' condition by a 'master Receiver'

If a 'master receiver' wants to conclude a data transfer, it must instruct the 'slave transmitter' to free data bus line SDA so that it can generate the 'stop' condition. The 'master receiver' therefore generates a negative acknowledge (data line SDA held HIGH) after reception of the current data byte. The 'slave transmitter' recognises the negative acknowledge by testing the LRB flag and must then be changed to the 'slave receiver' mode by software. The generation of the 'stop' condition consists of the following three stages:

1. An 8-bit byte is received without generating an acknowledgement bit because bit position ACK in clock control register S2 is loaded with a 0.
2. The 'master receiver' generates a negative acknowledgement by generating an extra clock pulse. i.e. it generates a single HIGH-state bit on data line SDA.
3. The 'stop' condition is generated as a 'master transmitter'.

The 'master receiver' program for the last received byte is the same as that given in section 4 until TNBY. After a check has verified that the last byte is to be transferred, the program continues with:

```
        MOV S2,#H'Ø3'   NO ACK, FSCLK ≃ 87 kHz
        MOV A,SØ        READ SØ AND START
¥                       WITH LAST BYTE TRANSFER
        MOV @RØ,A       STORE RECEIVED DATA
        MOV A,Rp        RESTORE CONTENTS OF ACCU
        RETR            RETURN TO MAIN PROGRAM
```

During the next serial interrupt routine, the program in section 4 is again followed until TNBY. After a check has verified that the last byte has been transferred, a negative acknowledge and a 'stop' condition must be transmitted. The program continues with:

```
        MOV S1,#H'A9'    LOAD BIT COUNTER FOR
¥                        ONE BIT TRANSFER
        MOV A,SØ         READ SØ AND START
¥                        ONE BIT TRANSFER =
¥                        NEGATIVE ACKNOWLEDGE
        MOV @RØ,A        STORE RECEIVED DATA
TSPI    MOV A,S1         LOAD STATUS WORD TO ACCU **
        JB4 TSPI         TEST END OF ONE BIT TRANSFER **
        MOV S2,#H'43'    RESTORE ACK BIT IN S2
        MOV S1,#H'D8'    GENERATE STOP CONDITION
        MOV A,Rp         RESTORE ACCU
        RETR             RETURN TO MAIN PROGRAM
```

** These instructions can be omitted if it is certain that the negative acknowledge has been transferred before instruction MOV S2, #43H is given.

8. Flow charts for the 'master' and 'slave' functions of the SIO interface

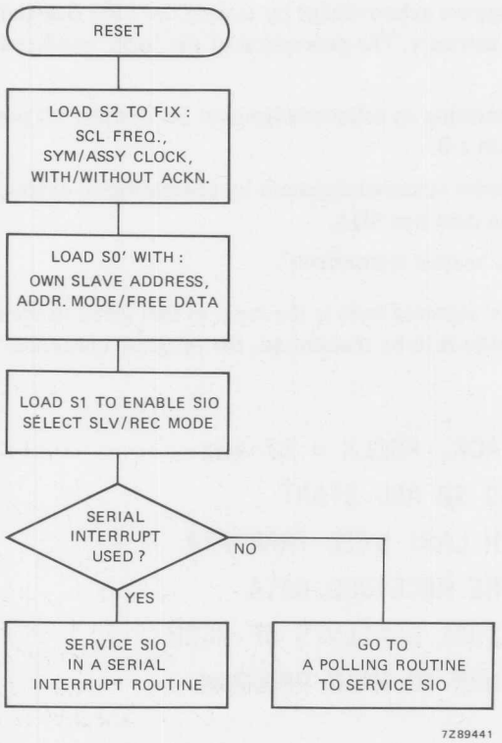


Fig. 11 Initialization flow chart. A prior RESET signal has cleared all bits of the SIO registers except PIN = 1 in status register S1.

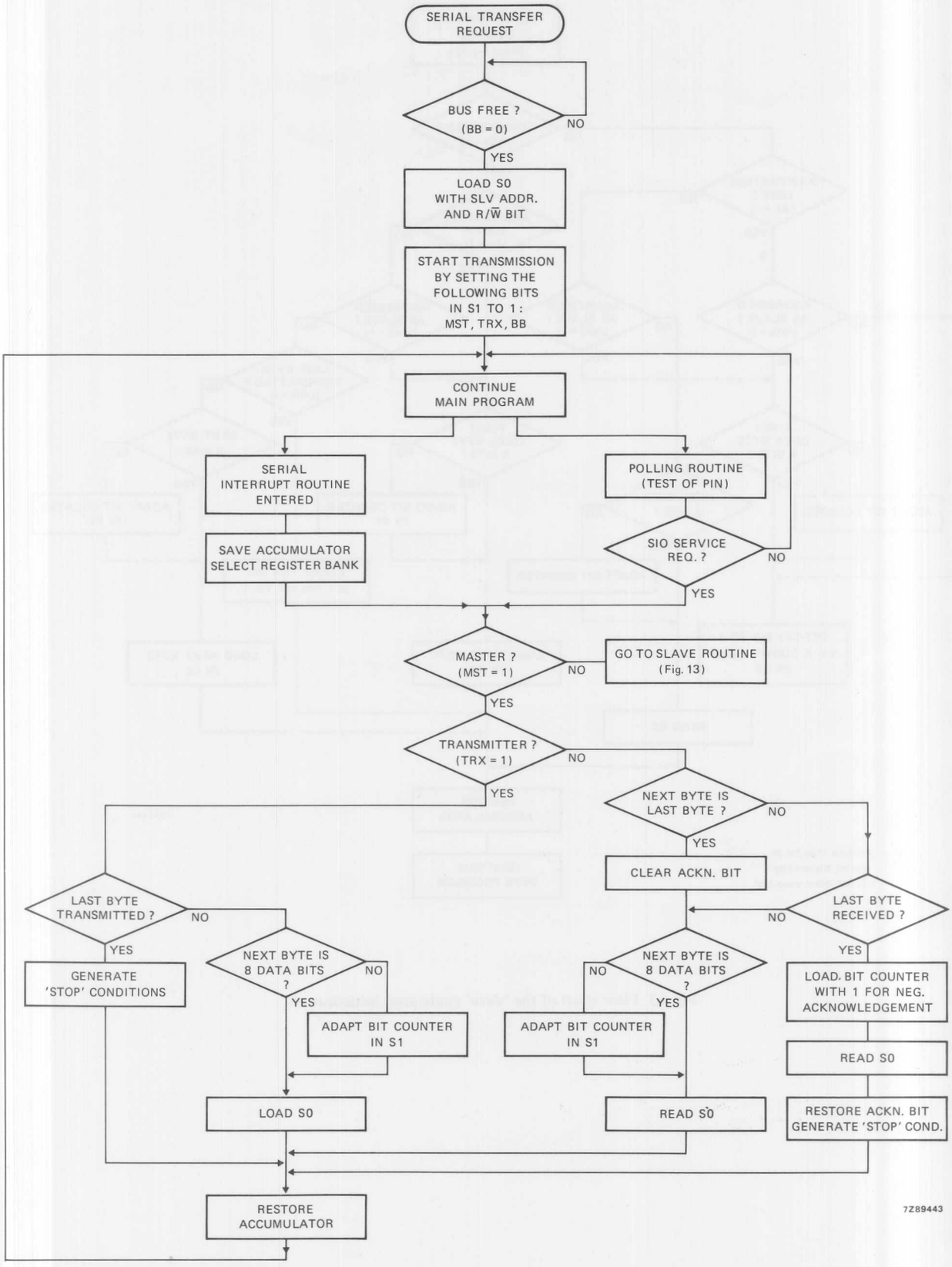


Fig. 12 Flow chart of the 'master' mode after initialization.

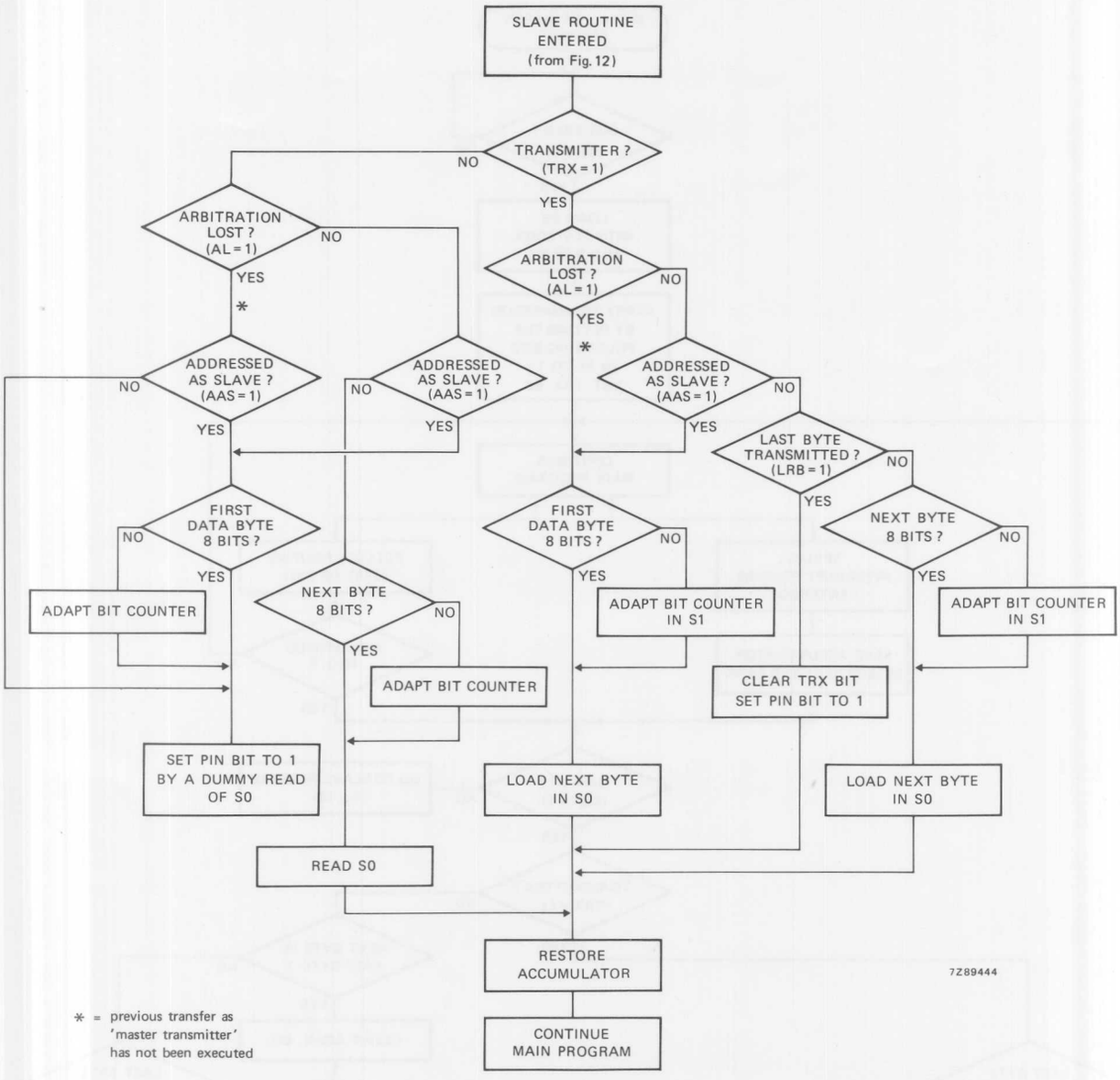


Fig. 13 Flow chart of the 'slave' mode after initialization.